

Studium wykonalności

3@KASK

31 marca 2009

Symbol projektu: 3@KASK	Opiekun projektu: mgr inż. Tomasz Boiński
Nazwa Projektu: Wizualizacja grafów za pomocą biblioteki Prefuse	

Nazwa Dokumentu: Studium wykonalności	Nr wersji: 0.0
Odpowiedzialny za dokument: Anna Jaworska	Data pierwszego sporządzenia: 31.03.09
Przeznaczenie: WEWNĘTRZNE	Data ostatniej aktualizacji: 31.03.09

Historia dokumentu

Wersja	Opis modyfikacji	Rozdział/strona	Autor modyfikacji	Data
1	Przygotowanie zarysu dokumentu i określenie zakresu badań	wszystkie	Anna Jaworska	31.03.09

Spis treści

1	Założenia realizacji studium	3
1.1	Podstawa wykonania i temat studium	3
1.2	Cel studium	3
1.3	Ograniczenia	3
2	Stan istniejący	3
2.1	Inne systemy i zasoby mające wpływ lub będące pod wpływem planowanego produktu . . .	3
2.2	Istniejące na rynku podobne rozwiązania	3
2.3	Problem i motywacja wdrożenia nowego produktu	3
3	Ogólne wymagania stawiane produktowi i ich priorytety	3
3.1	Architektura	3
3.2	Użytkownicy	3
3.3	Dane	4
3.4	Funkcjonalność	4
3.5	Wymogi techniczno - technologiczne	4
4	Ogólna ocena ryzyka i planowany sposób zarządzania nim	5
5	Uwarunkowania prawne i inne	5
6	Proponowane rozwiązania	5
7	Porównanie wariantów i rekomendacja	5
8	Strategia i wstępny harmonogram	5
	Literatura	6

1 Założenia realizacji studium

1.1 Podstawa wykonania i temat studium

Studium wykonywane jest przede wszystkim aby określić możliwe sposoby realizacji projektu. Ma także za zadanie zebranie i podsumowanie informacji potrzebnych zespołowi do realizacji projektu.

1.2 Cel studium

Celem studium jest zbadanie na potrzeby projektu *Wizualizacja grafów za pomocą biblioteki Prefuse*:

- jak należy tworzyć biblioteki w technologii JAVA
- jakich mechanizmów wizualizacji grafów dostarczają biblioteki JAVA
- czy realizacja projektu za pomocą Prefuse jest odpowiednim rozwiązaniem
- jaki standard OWL powinien być wspierany przez wytworzony produkt

1.3 Ograniczenia

Do podstawowych ograniczeń należą:

- konieczność realizacji projektu w języku JAVA
- konieczność wykorzystania wersji bibliotek zgodnych z użytymi w OCS
- limit czasowy projektu

2 Stan istniejący

2.1 Inne systemy i zasoby mające wpływ lub będące pod wpływem planowanego produktu

- OCS - Ontology C.. System
- OWL API ver 2.1.1 - API do przetwarzania plików w formacie OWL zgodnych ze standardem W3C; ta wersja API została użyta w projekcie OCS
-

2.2 Istniejące na rynku podobne rozwiązania

-

2.3 Problem i motywacja wdrożenia nowego produktu

Nowa biblioteka powinna powstać aby:

- ułatwić programistą wizualizację ontologii

3 Ogólne wymagania stawiane produktowi i ich priorytety

Wymienione wymagania mają charakter orientacyjny, pozwalający nakreślić zakres problemu jaki ma pokrywać projekt. Szczegółową definicję wymagań zawiera dokument *Specyfikacji wymagań*

3.1 Architektura

3.2 Użytkownicy

Użytkownikami biblioteki będą programiści tworzący aplikacje wizualizujące ontologie. Inicjalnie będą to programiści związani z projektem OCS, później mogą to być dowolni inni programiści chętni do korzystania z biblioteki.

3.3 Dane

Biblioteka powinna obsługiwać te same formaty danych co OWL API (zgodne ze specyfikacją W3C):

- RDF
- OWL Lite
- OWL DL
- OWL Full

Ponadto dane te powinny być wczytywane poprzez:

- podanie ścieżki do pliku OWL na dysku
- podanie adresu sieciowego zasobu z plikiem OWL
- podanie strumienia/kontenera z XML

3.4 Funkcjonalność

Zakładamy, że biblioteka będzie zawierać następujące funkcjonalności:

- wizualizacja elementów OWL
- pozwalać użytkownikowi na definiowanie akcji dla zdarzeń okna
- zawierać standardowe definicje zdarzeń
-

3.5 Wymogi techniczno - technologiczne

Produkt projektu, jako biblioteka w języku Java, powinien posiadać następujące cechy:

1. **Odpowiednie kapsułkowanie.** Publiczne powinny być jedynie te klasy i metody, które są istotne dla użytkownika i z których będzie on bezpośrednio korzystał.
2. **Możliwość debugowania.** Użytkownik powinien mieć możliwość debugowania kodu biblioteki, bez konieczności znajomości każdego jej szczegółu.
3. **Przejrzystość.** Kod biblioteki powinien być odpowiednio udokumentowany za pomocą javadoc. W szczególności, bardzo dokładnie należy opisać klasy oraz metody publiczne.
4. **Łatwość użycia.** Biblioteka powinna zawierać klasy, pokazujące przykłady wykorzystania jej klas i metod.
5. **Rozszerzalność.** Struktura wewnętrzna biblioteki powinna być odpowiednio podzielona na klasy (wykorzystując klasy abstrakcyjne i interfejsy). Dzięki temu użytkownik będzie miał możliwość stworzenia własnych klas, rozszerzających funkcjonalność biblioteki.
6. **Uniwersalność.** Biblioteka powinna mieć jasno określony problem, który rozwiązuje. Wyniki powinny być podane użytkownikowi w wygodny dla niego sposób (lub na kilka sposobów), który będzie umożliwiał wykorzystanie biblioteki w różnych aplikacjach. Innymi słowy, biblioteka powinna udostępniać łatwy i przejrzysty dla użytkownika interfejs.
7. Biblioteka powinna być napisana w taki sposób, aby użytkownik spojrzawszy na nią i mógł powiedzieć: *Wow, to jest dokładnie to, czego potrzebuję i dokładnie tak samo bym to napisał! ;).*

4 Ogólna ocena ryzyka i planowany sposób zarządzania nim

Schemat opisu czynnika ryzyka

ID czynnika	RISKXX
Nazwa czynnika	Nazwa
Opis czynnika	Opis...
Sposób zarządzania	Opis..

ID czynnika	RISK01
Nazwa czynnika	Problemy logistyczne zespołu
Opis czynnika	Uwzględniamy możliwość wystąpienia problemów osobistych członków zespołu powodujących ich wyłączenie z prac.
Sposób zarządzania	Jeśli ktoś zostanie wyłączony z prac, reszta zespołu musi podzielić między siebie jego obowiązki i informować osobę wyłączaną o postępie, tak aby miała wgląd w archiwa.

ID czynnika	RISK02
Nazwa czynnika	Problemy członków zespołu na uczelni
Opis czynnika	Możliwe jest powstanie zaległości związanych z innymi uczelnianymi obowiązkami
Sposób zarządzania	Członek zespołu musi zgłosić swoje problemy reszcie zespołu. W zależności od sytuacji termin wykonania jego zadań zostanie przedłużony lub zadania te przejmie ktoś inny.

ID czynnika	RISK03
Nazwa czynnika	Niedostępność opiekuna/klienta
Opis czynnika	Z różnych przyczyn niezależnych od zespołu opiekun może stać się niedostępny.
Sposób zarządzania	Wszelkie problemy wymagające według zespołu poznania opinii opiekuna będą musiały zostać rozwiązane poprzez podjęcie decyzji przez zespół bez wsparcia. Wszelkie problemy organizacyjne związane z projektem grupowym powinny pod nieobecność zgłaszane do katedralnego koordynatora projektów grupowych.

ID czynnika	RISKXX
Nazwa czynnika	Nazwa
Opis czynnika	Opis...
Sposób zarządzania	Opis..

ID czynnika	RISKXX
Nazwa czynnika	Nazwa
Opis czynnika	Opis...
Sposób zarządzania	Opis..

ID czynnika	RISKXX
Nazwa czynnika	Nazwa
Opis czynnika	Opis...
Sposób zarządzania	Opis..

5 Uwarunkowania prawne i inne

6 Proponowane rozwiązania

7 Porównanie wariantów i rekomendacja

8 Strategia i wstępny harmonogram

Literatura